

# Matrix eQTL: Ultra fast eQTL analysis via large matrix operations

Andrey A. Shabalin

Department of Biostatistics, University of North Carolina at Chapel Hill

## Abstract.

**Motivation:** Expression quantitative trait loci (eQTL) mapping aims to determine genomic regions that regulate gene transcription. Expression QTL is used to study the regulatory structure of normal tissues and to search for genetic factors in complex diseases such as cancer, diabetes, and cystic fibrosis.

A modern eQTL dataset contains millions of SNPs and thousands of transcripts measured for hundreds of samples. This makes the analysis computationally complex as it involves independent testing for association for every transcript-SNP pair. The heavy computational burden makes eQTL analysis less popular, often forces analysts to restrict their attention to just a subset of transcripts and SNPs. As larger genotype and gene expression datasets become available, the demand for fast tools for eQTL analysis increases.

**Solution:** We present a new method for fast eQTL analysis via linear models, called Matrix eQTL. Matrix eQTL can model and test for association using both linear regression and ANOVA models. The models can include covariates to account for such factors as population structure, gender, and clinical variables. It also supports testing of heteroscedastic models and models with correlated errors. In our experiment on large datasets Matrix eQTL was thousands of times faster than the existing popular software for QTL/eQTL analysis.

Matrix eQTL is implemented as both Matlab and R packages and thus can easily be run on Windows, Mac OS, and Linux systems. The software is freely available at the following address.

**Website:** [http://www.bios.unc.edu/research/genomic\\_software/Matrix\\_eQTL](http://www.bios.unc.edu/research/genomic_software/Matrix_eQTL)

## Author Summary.

Expression quantitative trait loci (eQTL) analysis is a part of genetical genomics linking variations in gene expression levels to the differences in genotype. It is important for understanding of the regulatory functioning in both normal and diseased tissues. Modern eQTL studies measure gene expression over tens of thousands of transcripts and variations of genotype (single nucleotide polymorphisms, SNPs) over millions of markers. This draws the analysis extremely computationally intensive as it involves independent testing for association for every transcript-SNP pair. Computational burden makes eQTL analysis less popular, often forces analysts to restrict their attention to a small subset of genes-SNP pairs.

We present a new fast tool for eQTL analysis called Matrix eQTL. It is designed to handle large datasets and is hundreds to thousands times faster than the existing tools for eQTL analysis. Using Matrix eQTL analysts can fit and compare multiple models in a matter of minutes or hours, even for large datasets. Matrix eQTL makes it fast and simple to test different preprocessing and quality control procedures. One can also use Matrix eQTL to determine the right significance thresholds via permutation testing. Matrix eQTL will greatly improve the ability of analysts to find true eQTLs by fitting the right model and determining the correct significance threshold.

## Introduction.

The goal of eQTL analysis is to identify genomic locations where the genotype is significantly associated with expression of known genes. These associations can help reveal biochemical process underlying living systems, discover the genetic factors causing certain diseases and determine pathways that are affected by them. Expression QTL analysis is used to determine hotspots (DNA

regions affecting multiple transcripts, [Breitling et al. \[2008\]](#)), construct causal networks, discover subclasses of clinical phenotypes and select genes for clinical trials [see reviews of [Gilad et al., 2008](#), [Zhang and Liu, 2010](#), [Kendzierski and Wang, 2006](#)].

There are various approaches to eQTL analysis. Some methods test for transcript-SNP associations independently, while others aim to find multiple SNPs that can jointly explain variations in expression of a gene. There is a wide range of models used to fit the expression values. In this paper our goal is fast analysis of large datasets and thus we focus on independent testing of each transcript-SNP pair using linear regression. More elaborate approaches include non-linear modeling, such as generalized linear models, Bayesian models, and models accounting for pedigree.

Expression QTL analysis is known to be computationally intensive. The issue is most pronounced for modern eQTL dataset, which often contain millions of SNPs and thousands of transcripts measured for hundreds of samples. For such data, the complete eQTL analysis may involve over ten billion tests. The following three studies indicate that non-linear methods can be prohibitively slow for large datasets. First, [Degnan et al. \[2008\]](#) were applying Family Based Association Tests (FBAT) to a dataset with 142 samples. They tested only 40,227 transcript-SNP pairs and report the total computation time to be ‘under 24 hours on 20 processors in parallel on a Linux cluster’. Second, [Ghazalpour et al. \[2008\]](#) were running Efficient Mixed-Model Association (EMMA), which is claimed to be computationally efficient. They analyzed a dataset with 110 samples, 1,813 SNPs, and 10,013 transcripts. They tested all transcript-SNP pairs and report the computation time of ‘a few hours using a cluster of 50 processors’. Third, [Listgarten et al. \[2010\]](#) tested their method on a dataset containing 40,639 probes in the expression data and 48,186 SNPs for 188 samples. They report that estimation took 10 hours when parallelized across 1,100 processors (this is more than a processor-year). Note that the dimensions of a modern eQTL dataset can greatly exceed those in the examples above.

For a large dataset, eQTL analysis using existing tools requires multiple processor-weeks, and thus often has to be performed on a computing cluster. In this paper we present a new ultra fast method, called Matrix eQTL, for eQTL analysis using linear models. Matrix eQTL allows to analyze large data on a single desktop machine in the time existing tools would run on a large computing cluster. In our tests on a cystic fibrosis dataset with 840 samples Matrix eQTL strongly outperformed existing QTL/eQTL tools.

Matrix eQTL performs separate testing for each transcript-SNP pair. The association between each SNP and each transcript is tested using either least squares regression or ANOVA model. Both linear regression and ANOVA models can include extra additive covariates to account for such factors as population structure, gender, and clinical variables. Matrix eQTL also supports such extensions of linear regression model as weighted least squares, model with correlated errors, and mixed effects model with known variance parameters.

Matrix eQTL uses a special algorithm and data preprocessing allowing testing for the association between the genotype and expression without estimation of all the model parameters. The most computationally intensive part of the algorithm is formulated in terms of operations with large matrices. This allowed us to implement the fast algorithm using high-level programming languages, R and Matlab, relying on their efficient implementation of matrix operations. The performance of Matrix eQTL depends on the efficiency of matrix multiplication routine. More information about matrix multiplication and its implementations in R and Matlab is provided in the Appendix.

Matrix eQTL is cross platform and can be run on any platform for which Matlab or R is available, namely Linux, Mac OS X, and Windows. The Matlab and R implementations of Matrix eQTL are independent and have equal functionality. The performance of the implementations may differ depending on the version of Matlab or R used to run the code.

## Results.

**Data.** To assess the performance of Matrix eQTL and compare it to other eQTL tools we used genotype and gene expression data from 840 patients with cystic fibrosis [Wright et al. \[2011\]](#). The genotype information was obtained for 573,337 markers and the gene expression was measured for 22,011 transcripts. The missing values were imputed by average values of the variable across samples.

**Performance.** We compare performance of Matrix eQTL with that of five programs for QTL and eQTL analysis: Plink [[Purcell et al., 2007](#)], Merlin [[Abecasis et al., 2001](#)], R/qtl [[Broman et al., 2003](#)], eMap [[Sun, 2009](#)], and FastMap [Gatti et al. \[2009, 2011\]](#). Plink is a command line toolset for whole genome association analysis. It is written in C/C++ and as a general purpose tool is not optimized for eQTL analysis. Merlin is a command line tool for fast pedigree analysis written in C/C++. It is designed for analysis in pedigree and may not show the best performance for unrelated samples. R/qtl and eMap are R packages for QTL/eQTL analysis, part of eMap is coded in C and requires GSL library (GNU Scientific Library). FastMap is a user friendly tool for fast association mapping written in Java. It uses discrete nature of genotype data to speed up calculations, but it can not handle covariates. FastMap is optimized for permutation based testing. Note that FastMap is the only tool out of the five that has a graphical user interface.

All methods except R/qtl were set to estimate the simple linear model for the relationship between gene expression and genotype. R/qtl does support the simple linear model and was set to estimate the ANOVA model (Haley-Knott method). First, we ran all methods on a subset of the cystic fibrosis data, containing 2,000 random genes and SNPs. To reduce the output we set, where possible, the p-value threshold at  $10^{-5}$  level. The technical specifications of the machine used for testing are provided in the Appendix. Table 1 shows that the first four existing methods performed the analysis in more than eight minutes and FastMap finished in about one minute, while both versions of Matrix eQTL (Matlab and R) finished in less than half of a second.

The time required for the analysis of the complete datasets is presented in the right column of the table. For the five existing methods the time is estimated under the assumption of linearity with respect to the number of transcripts and the number of SNPs. Both implementations of matrix eQTL were applied to the complete dataset to obtain the precise timing.

Method	2k Genes 2k SNPs	Complete dataset
Plink	2678 sec	97.8 days
Merlin	564 sec	20.6 days
R/qtl	596 sec	21.2 days
eMap	492 sec	18.0 days
FastMap	61 sec	2.2 days
Matrix eQTL (Matlab)	0.23 sec	11.5 minutes
Matrix eQTL (Rev R)	0.47 sec	13.4 minutes

Table 1: Performance of various eQTL software on the Cystic Fibrosis dataset. The time for first 4 methods is projected from a random subset of 2,000 genes and 2,000 SNPs.

Matrix eQTL can perform analyses that previously required days or weeks in just minutes. QTL modelers that have only a handful of phenotypes understand the importance of testing several different models with covariates and interaction terms. Until now, this has not been computationally feasible with eQTL analyses due to the computational burden. Matrix eQTL allows analysts to fit

a variety of models with different mixes of covariates in less than an hour and compare the results. Likewise, it allows analysts to compare different preprocessing and quality control procedures. Furthermore, once an appropriate model has been selected, Matrix eQTL can be used to determine permutation based significance thresholds in less time than most packages take to generate nominal p-values. Matrix eQTL will greatly improve the ability of analysts to find true eQTLs by fitting the right model and determining the correct significance threshold.

## Methods.

We describe the algorithm of Matrix eQTL in steps. First, we detail the algorithm for the simple linear regression. Then we show how the algorithm is extended to handle the ANOVA model and covariates. Finally we show how the algorithm is extended to test heteroskedastic models and models with correlated errors.

**Simple linear regression.** The simple linear regression is probably the commonly used model for eQTL analysis. For each transcript-SNP pair, the association between gene expression  $g$  and genotype  $s$  is assumed to be linear, with genotype encoded as 0, 1, or 2 according to the frequency of the minor allele.

$$g = \alpha + \beta s + \epsilon, \quad \text{where } \epsilon \sim i.i.d. N(0, \sigma^2) \quad (1)$$

The conventional algorithm for the analysis of the simple linear regression includes estimation or calculation of a number of parameters: the sample means  $\bar{g}$  and  $\bar{s}$ , the slope coefficient  $\hat{\beta}$ , the intercept  $\hat{\alpha}$ , the residuals  $e_i$ , the total sum of squares  $SST$ , and the residual sum of squares  $SSE$ . This is followed by the estimation of a test statistic, which can be t-statistic, F-test, or the likelihood ratio test. Finally, the p-value is calculated for the test statistic; this step can also be computationally intensive as it involves calculation of incomplete beta or gamma functions.

The goal of Matrix eQTL is to find all transcript-SNP pairs with association significant at a given level. This allows us to skip estimation of unnecessary parameters and focus on the most efficient calculation of a test statistic.

To save time, Matrix eQTL does not calculate p-value for every transcript-SNP pair. Instead, for the test statistic of choice, it finds the threshold, above which the test statistic is significant at the required significance level. The test statistics for every transcript-SNP pair are then compared to the threshold, and the p-values are calculated only for those above the threshold.

The choice of test statistic is important performance of the method. It is natural to choose the test statistic that can be calculated faster, among statistics of equal power. Observe that for the simple linear regression (1), the common statistics, such as  $t$ ,  $F$ ,  $R^2$ , and  $LR$ , are equivalent and can be expressed as functions of the correlation  $r = \text{cor}(g, s)$

$$t = \sqrt{n-2} \frac{r}{\sqrt{1-r^2}}, \quad F = t^2 = (n-2) \frac{r^2}{1-r^2}, \quad R^2 = r^2, \quad LR = -\log(1-r^2).$$

Thus we choose the absolute value of the correlation  $|r|$  as the test statistic for the simple linear regression and threshold it to find significant transcript-SNP associations. Note that correlation does not change if we standardize the genotype and gene expression variables such that

$$\sum g_i = 0, \quad \sum g_i^2 = 1, \quad \sum s_i = 0, \quad \sum s_i^2 = 1.$$

The standardization does not add complexity to the calculations as it has to be performed only once for each transcript and once for each SNP, and it greatly simplifies the calculation of the correlation

$$r = \text{cor}(s, g) = \frac{\sum (s_i - \bar{s})(g_i - \bar{g})}{\sqrt{\sum (s_i - \bar{s})^2 \sum (g_i - \bar{g})^2}} = \sum s_i g_i = \langle s, g \rangle,$$

where  $\langle s, g \rangle$  denotes the inner product between vectors  $s$  and  $g$ . Now, let  $S$  be the genotype matrix, with each row containing measurements for a single SNP and each column containing measurements for a single sample. Let  $G$  be the gene expression matrix, with each row containing measurements for a single transcript and each column containing measurements for a single sample. Let the columns (samples) of matrix  $S$  match those of matrix  $G$ . Then the matrix of all gene-SNP correlations can be calculated in just one matrix multiplication  $GS^T$  as illustrated in Figure 1.

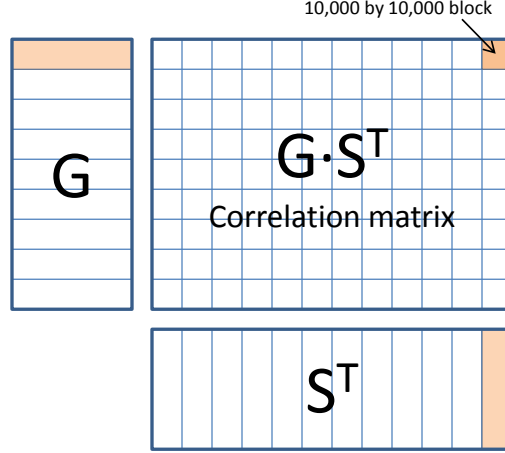


Figure 1: Matrix of correlations can be calculated using matrix multiplication. Due to the huge number of tests the analysis performed in 10,000x10,000 blocks.

However, the number of tests in a modern eQTL study may exceed tens of billions and such correlation matrix would require hundreds of gigabytes of RAM. To avoid excess memory requirements we slice the data matrices in blocks of up to 10,000 variables and perform the analysis separately for each pair of blocks. Figure 1 illustrates the slicing and calculation of the correlation matrix.

The algorithm of Matrix eQTL for the simple linear regression is as follows:

1. Split input matrices into blocks of up to 10,000 variables.
2. Standardize variables of both gene expression and genotype matrices.
3. For each pair of blocks:
  - (a) Calculate the corresponding block of the correlation matrix in one matrix operation
  - (b) Find absolute correlations exceeding predefined threshold
  - (c) For those correlations, calculate test statistic, p-value, and other variables of interest and report them

**Model with covariates.** It is common to include covariates in the eQTL model to account for such effects as population stratification, gender, age, and clinical variables. For simplicity, let's consider the model with one extra covariate  $x$ ,

$$g = \alpha + \gamma x + \beta s + \epsilon. \quad (2)$$

As for the simple linear regression, various test statistics testing for the significance of  $s$ , namely LR, F-test, and t-statistic, are equivalent. The testing can be reduced to the simple linear regression model by orthogonalization of  $g$  and  $s$  with respect to  $x$ . The algorithm for the analysis is then:

1. Center variables  $g$ ,  $x$ , and  $s$  to remove  $\alpha$  from the model.
2. Orthogonalize  $g$  and  $s$  with respect to  $x$ :  $\tilde{g} = g - \langle g, x \rangle x$ ,  $\tilde{s} = s - \langle s, x \rangle x$ .
3. Perform the analysis for the simple linear regression  $\tilde{g} = \beta \tilde{s} + e$  using one less degree of freedom for the test statistic to account for the removed covariate.

**ANOVA model.** Another common extension of the simple linear regression for eQTL analysis is to treat each genotype variable as categorical and model its effect on gene expression with ANOVA model. ANOVA model can be viewed as a linear regression with each SNP represented with two dummy variables:  $s_1 = I(s = 1)$  and  $s_2 = I(s = 2)$ .

$$g = \alpha + \beta_1 s_1 + \beta_2 s_2 + \epsilon \quad (3)$$

F-test or LR statistic are equivalent statistics for testing joint significance of  $s_1$  and  $s_2$ . Both F and LR are monotone functions of  $R^2$  for this model, thus we can use  $R^2$  as the test statistic. The test statistic can be calculated efficiently if we orthogonalize the regressors. The algorithm for the analysis is then:

1. Center variables  $g$ ,  $s_1$ , and  $s_2$  to remove  $\alpha$  from the model.
2. Orthogonalize  $s_2$  with respect to  $s_1$  for every marker,  $\tilde{s}_2 = s_2 - \langle s_2, s_1 \rangle s_1$ , and standardize it.
3. Use test statistic:  $R^2 = \langle g, s \rangle^2 + \langle g, \tilde{s}_2 \rangle^2$
4. The threshold for  $R^2$  and p-values can be derived from the formula for F-test  $F = \frac{(n-3)R^2}{2(1-R^2)}$ .

The same algorithm can be used to estimate the model with two marker-by-marker variables, such as genotype and copy number variations.

Another test statistics can be used to test for the significance of just one variable ( $s_2$ ) accounting for the effect of another marker-by-marker variable ( $s_1$ ). In this case the test statistic would be

$$F = \frac{n-3}{2} \cdot \frac{\langle g, \tilde{s}_2 \rangle^2}{1 - \langle g, s_1 \rangle^2 - \langle g, \tilde{s}_2 \rangle^2}.$$

This approach can easily be generalized for testing for joint significance of any subset of regressors.

**Heteroskedastic models and models with correlated errors.** The previously described models assume the noise to be independent and identically distributed across samples. However, the errors can be heteroskedastic if the quality of the measurements differs across samples. Also, the errors may be correlated if the samples come from a pedigree. To account for both possibilities, consider the model with non i.i.d. errors:

$$g = \alpha + \beta s + u, \quad \text{where } U \sim N(0, \sigma^2 K) \quad (4)$$

and  $K$  is a known non-singular covariance matrix. To apply the previously described methods to this problem we transform the input variables to make the errors independent and identically distributed:

$$\tilde{g} = K^{-1/2}g, \quad \tilde{s} = K^{-1/2}s, \quad \tilde{q} = K^{-1/2}1_n,$$

where  $1_n$  is a vector of ones. The new model equation is homoskedastic, has independent errors, but does not include a constant.

$$\tilde{g} = \tilde{q} + \beta \tilde{s} + e, \quad \text{where } e \sim i.i.d. N(0, \sigma^2) \quad (5)$$

The model is tested using the algorithm for the linear model with covariates with step 1 (centering) omitted.

## Grant Support

This work was supported, in part, by funding from the National Institutes of Health (R01-MH090936 and R01-ES015241), US Environmental Protection Agency (STAR RD83382501 and RD83272001), National Cancer Institute (R01-CA138255), National Institute of Mental Health (R01-MH090936), and the Gillings Innovation Laboratory in Statistical Genomics. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

## Conclusion

In this paper we presented a new ultra fast method for eQTL analysis, called Matrix eQTL. The method can test for transcript-SNP associations using linear regression and ANOVA models with covariates. It also supports models with heteroskedastic and correlated errors to account for sample quality and population structure. We tested Matrix eQTL and compared it to five existing eQTL methods. Matrix eQTL was hundreds to thousands of times faster than the other eQTL methods. Matrix eQTL allows one to perform eQTL analysis of large datasets on a single desktop machine in the time other methods would run on a large computing cluster. Matrix eQTL is implemented in Matlab and R, and the code is publicly available.

Fast performance of Matrix eQTL opens new possibilities for analysts. Matrix eQTL does not require a computing cluster. Using Matrix eQTL analysts can fit and compare multiple models in a matter of minutes or hours, even for large datasets. Matrix eQTL simplifies testing of different preprocessing and quality control procedures. Matrix eQTL can also be used to determine significance thresholds via permutation testing. Matrix eQTL will greatly improve the ability of analysts to find true eQTLs by fitting the right model and determining the correct significance threshold.

For the future versions of Matrix eQTL we plan to consider several extensions and modifications. First, we consider performing calculation on a GPU (graphics processing unit), instead of the CPU. We estimate that the GPU version of Matrix eQTL would provide at least tenfold increase in the performance (see Appendix). We will also consider using Matrix eQTL optimization techniques for fast estimation of more complex models, such as multi-SNP models and generalized linear model.

## Appendix.

**Matrix Multiplication.** Matrix eQTL gains its efficiency by expressing the most computationally intensive part of the analysis in terms of large matrix operations, most importantly matrix multiplication. Naturally, the performance of Matrix eQTL depends strongly on the performance of the employed basic linear algebra subroutine (BLAS). In Table 2 we compare performance of matrix multiplication for Matlab and different versions of R for Windows. For the comparison we measured the time required to multiply two 4,096x4,096 matrices with elements set to random values uniformly distributed on [0,1]. The standard installation of R for Windows [R Development Core Team, 2010] includes a generic BLAS, not optimized for any particular CPU. The test finished in 90 and 80 seconds for 32 and 64 bit versions of R respectively. There is a faster version of BLAS available for R (32 bit only) called ATLAS [Whaley and Petitet, 2005]. The matrix multiplication test on R with C2D version of ATLAS library finished in just 15 seconds. Next we tested Matlab and Revolution R, a commercial version of R available free of charge for academic purposes. They both employ Intel Kernel Math Library (KML), which is optimized for Intel CPUs (as in the test machine) and is able to use multiple CPU cores. The test for both programs finished in 4.3 seconds. Intel KML uses all 4 cores of the test machine and demonstrates about 4 times better performance than R with ATLAS library. About twice better performance can be achieved by switching from



double to single precision calculations; we did not use this option in Matrix eQTL to avoid loss of accuracy. Single precision calculation are not available in R. The last line in the table shows that NVIDIA GTX 480 GPU (graphics processing unit) can offer ten times better performance than the best algorithm for the CPU (central processing unit) of the test machine. Matlab 2010b has limited build-in support for GPU-based calculations.

The complexity of the direct matrix multiplication for square  $n \times n$  matrices is  $O(n^3)$ . More asymptotically efficient methods have been developed with complexity  $O(n^{\log_2 7}) \approx O(n^{2.81})$  [Strassen, 1969] and even  $o(n^{2.376})$  [Coppersmith and Winograd, 1990]. However, in practice, the new methods do not beat the direct one even for relatively large matrices ( $n \approx 2000$ ), and in certain circumstances they may experience numerical instability.

Package	BLAS	time (sec)	comment
R x32 2.12.1	build-in	90	
R x64 2.12.1	build-in	80	
R x32	Atlas C2D	15	
Revolution R 4.0	Intel KML	4.3	
Matlab R2010b	Intel KML	4.3	
Matlab R2010b	Intel KML	2.2	single precision
Matlab R2010b	GPU CUDA	0.25	GTX 480, single precision

Table 2: Performance of different software in multiplying 4,096x4,096 matrices.

**Specifications of the computer and software used for testing:** Brand: Lenovo ThinkStation E20; CPU: Intel Xeon X3430 (2.4 ghz, 4 cores, 38.4 gflop); RAM: 16 GB DDR3; OS: Windows 7; Matlab R2010b; Revoluton R Enterprise 4.0 (64 bit).

## References

- G.R. Abecasis, S.S. Cherny, W.O. Cookson, and L.R. Cardon. Merlin—rapid analysis of dense genetic maps using sparse gene flow trees. *Nature genetics*, 30(1):97–101, 2001.
- R. Breitling, Y. Li, B.M. Tesson, J. Fu, C. Wu, T. Wiltshire, A. Gerrits, L.V. Bystrykh, G. De Haan, A.I. Su, et al. Genetical genomics: spotlight on QTL hotspots. *PLoS Genet*, 4(10):e1000232, 2008.
- K.W. Broman, H. Wu, S. Sen, and G.A. Churchill. R/qtl: QTL mapping in experimental crosses. *Bioinformatics*, 19(7):889, 2003. ISSN 1367-4803.
- D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *Journal of symbolic computation*, 9(3):251–280, 1990. ISSN 0747-7171.
- J.H. Degnan, J. Lasky-Su, B.A. Raby, M. Xu, C. Molony, E.E. Schadt, and C. Lange. Genomics and genome-wide association studies: An integrative approach to expression QTL mapping. *Genomics*, 92(3):129–133, 2008. ISSN 0888-7543.
- D.M. Gatti, A.A. Shabalina, T.C. Lam, F.A. Wright, I. Rusyn, and A.B. Nobel. FastMap: Fast eQTL mapping in homozygous populations. *Bioinformatics*, 25(4):482, 2009. ISSN 1367-4803.
- D.M. Gatti, A.A. Shabalina, M Sypa, T.C. Lam, F.A. Wright, I. Rusyn, and A.B. Nobel. FastMap 2.0: Fast association mapping in heterozygous populations. *Working paper*, 2011.



- A. Ghazalpour, S. Doss, H. Kang, C. Farber, P.Z. Wen, A. Brozell, R. Castellanos, E. Eskin, D.J. Smith, T.A. Drake, and L.J. Aldons. High-resolution mapping of gene expression using association in an outbred mouse stock. *PLoS Genet*, 4(8):e1000149, 08 2008. doi: 10.1371/journal.pgen.1000149. URL <http://dx.doi.org/10.1371%2Fjournal.pgen.1000149>.
- Y. Gilad, S.A. Rifkin, and J.K. Pritchard. Revealing the architecture of gene regulation: the promise of eqtl studies. *Trends in genetics*, 24(8):408–415, 2008.
- C. Kendzierski and P. Wang. A review of statistical methods for expression quantitative trait loci mapping. *Mammalian genome*, 17(6):509–517, 2006. ISSN 0938-8990.
- J. Listgarten, C. Kadie, E.E. Schadt, and D. Heckerman. Correction for hidden confounders in the genetic analysis of gene expression. *Proceedings of the National Academy of Sciences*, 107(38):16465, 2010. ISSN 0027-8424.
- S. Purcell, B. Neale, K. Todd-Brown, L. Thomas, M.A.R. Ferreira, D. Bender, J. Maller, P. Sklar, P.I.W. De Bakker, M.J. Daly, et al. PLINK: a tool set for whole-genome association and population-based linkage analyses. *The American Journal of Human Genetics*, 81(3):559–575, 2007. ISSN 0002-9297.
- R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2010. URL <http://www.R-project.org>. ISBN 3-900051-07-0.
- V. Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13(4):354–356, 1969. ISSN 0029-599X.
- W. Sun. eQTL Analysis by Linear Model. 2009.
- R.C. Whaley and A. Petitet. Minimizing development and maintenance costs in supporting persistently optimized BLAS. *Software: Practice and Experience*, 35(2):101–121, 2005. ISSN 1097-024X.
- F.A. Wright, L.J. Strug, V.K. Doshi, C.W. Commander, S.M. Blackman, L. Sun, Y. Berthiaume, D. Cutler, A. Cojocaru, J.M. Collaco, M. Corey, R. Dorfman, K. Goddard, D. Green, J.W. Kent Jr, E.M. Lange, S. Lee, W. Li, J. Luo, G.M. Mayhew, K.M. Naughton, R.G. Pace, P. Pare, J.M. Rommens, A. Sandford, J.R. Stonebraker, W. Sun, C. Taylor, L.L. Vanscoy, F. Zou, J. Blangero, J. Zielenski, W.K. O’Neal, M.L. Drumm, P.R. Durie, M.R. Knowles, and G.R. Cutting. Genome-wide association and linkage identify modifier loci of lung disease severity in cystic fibrosis at 11p13 and 20q13.2. *Nature Genetics*, 43(6):539–546, 2011.
- W. Zhang and J.S. Liu. From QTL Mapping to eQTL Analysis. *Frontiers in Computational and Systems Biology*, pages 301–329, 2010.